1.2 No 3 et 4 : Loi de Gauss-Laplace et loi de Gauss-Laplace inverse

$$\mathbb{P}(X < x) = \frac{1}{2\pi} \int_{-\infty}^{x} \exp^{-t^2/2}\ dt$$

est la probabilité pour qu'une variable aléatoire X qui suit la loi normale centrée réduite soit inférieure à une valeur x. On propose de calculer $\mathbb{P}(x)$ numériquement ; puis, toujours numériquement, de résoudre par rapport à x l'équation $\mathbb{P}(x) = p$.

Question 1 Est-il possible de calculer une intégrale généralisée (i.e. dont l'une des bornes est infinie) par une méthode numérique ?

Non, sauf par changement de variable

Question 2 Montrer que $\int_{-\infty}^{+\infty} \exp^{-t^2/2}\ dt = \sqrt{2\,\pi}$ et en déduire que le calcul de $f(x) = \int_0^x \exp^{-t^2/2}\ dt$ suffit pour obtenir celui de $\mathbb{P}(X < x)$.

$$\begin{aligned}
\int_{-\infty}^{+\infty} \exp^{-t^2/2}\ dt &= \sqrt{\int_{-\infty}^{+\infty} \exp^{-x^2/2}\ dx\ \int_{-\infty}^{+\infty} \exp^{-y^2/2}\ dy} \\
&= \sqrt{\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \exp^{-x^2/2} \exp^{-y^2/2}\ dxdy} \\
&= \sqrt{\int_0^{2\pi} d\theta \int_0^{\infty} rdrexp^{-r^2/2}} \\
&= \sqrt{2\pi}
\end{aligned}$$

On écrit $\mathbb{P}(X < x) = \frac{1}{2} + \frac{1}{\sqrt{2\pi}} \underbrace{\int_0^x \exp^{-t^2/2}\ dt}_{= f(x)}$

Question 3 En utilisant les tableaux donnés, calculer $f(1.96)$, dont la valeur exacte est 1.190653... (où on a donné les 6 premières décimales) avec la méthode de Gauss-Legendre d'ordre 1 ; d'ordre 3 ; d'ordre 5 et d'ordre 7.

à l'ordre	:	1	3	5	7
f(1.96)	=	1.212572	1.196163	1.190247	1.190669

Faire remarquer que si N est le nombre de point d'intégration, $2\,N-1$ est l'ordre, l'ordre étant le degré du polynôme qui est intégré exactement par la méthode. Faire attention également au fait qu'il y a un changement de variable à faire $x = 0.98 + 0.98\,t$ pour que $t \in [-1, 1] \Longrightarrow x \in [0, 1.96]$ de manière à calculer $0.98 \int_{-1}^{1} exp^{-(0.98+0.98\,t)^2/2}\ dt$ par Gauss-Legendre.

x_n	0
w_n	2
x_n	$\pm$ 0.57735026918963
w_n	1

x_n	0	$\pm$ 0.77459666924148
w_n	0.88888888888889	0.55555555555556
x_n	$\pm$ 0.86113631159405	$\pm$ 0.33998104358486
w_n	.34785484513745	0.65214515486255

Question 4 Écrire l'algorithme de calcul de $\mathbb{P}(x < X)$ en considérant qu'on dispose d'une fonction qui donne les points du support et les poids de Gauss-Laplace à n'importe quel ordre.

Données : x, n le nombre de points de Gauss-Laplace à utiliser
Résultat : $\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \exp^{-t^2/2} \ dt$
début
  setp $\longleftarrow$ support_et_poids_de_gauss_legendre(n);
  $Res \longleftarrow 0$;
  tant que $n>0$ **faire**
    $Res \longleftarrow Res + setp[n][2] \times exp^{-x^2} \times (1 + setp[n][1])^2/8$;
    $n \longleftarrow n-1$
  fin
  $Res \longleftarrow \frac{1}{2} + \frac{1}{\sqrt{2\pi}} \times Res \times \frac{x}{2}$
fin

Question 5 Voici la fonction écrite en langage de maxima qui réalise la tâche de la question précédente.

```
load(orthopoly);
support_et_poids_de_gauss_legendre(n):=
/* renvoie le tableau [[x1,w1],...,[xn,wn]] composés des tableaux [points de support,poids].
adaptation de gauleg--Rybicki du numerical recipes */
block([polynome,derivee,racines,poids,x,precision:10^(-15)],
polynome:legendre_p(n,x),derivee:diff(polynome,x),
racines:realroots(polynome,precision),
map(lambda([u],ev(float([x,2/((1-x^2)*derivee^2)]),u)),racines)
);
```

Utiliser cette fonction pour réaliser une fonction correspondant à l'algorithme de la question précédente.

```
P(x,n):=block([res:0,setp,ps],
setp:support_et_poids_de_gauss_legendre(n),
for i:1 step 1 thru n do (ps:setp[i],res:res+ps[2]*exp(-x^2/8*(1+ps[1])^2)),
float(1/2+res*x/2/sqrt(2*%pi)));
```

ou mieux

```
P(x,n):=block([res:0,setp,ps,x2],
if integerp(n) then setp:support_et_poids_de_gauss_legendre(n)
               else (setp:n,n:length(setp)),
x2:x^2/8,
for i:1 step 1 thru n do (ps:setp[i],res:res+ps[2]*exp(-x2*(1+ps[1])^2)),
float(1/2+res*x/2/sqrt(2*%pi)));
```

Question 6 On veut résoudre l'équation $\mathbb{P}(x) = y$ par rapport à x pour trouver la loi de Gauss-Laplace inverse. Écrire une approximation à l'ordre 1 de cette équation autour du point x_0

$\mathbb{P}(x) = \mathbb{P}(x_0 + (x - x_0)) = \mathbb{P}(x_0) + \frac{\exp^{-x_0^2/2}}{\sqrt{2\pi}}(x - x_0) + \ldots$ d'où l'approximation à l'ordre 1 de l'équation $\mathbb{P}(x_1) + \frac{\exp^{-x_0^2/2}}{\sqrt{2\pi}}(x_1 - x_0) = y$ à partir de laquelle on obtient

$$x_1 = x_0 + \sqrt{2\pi}\ exp^{x_0^2/2}(y - \mathbb{P}(x_0))$$

Question 8 En déduire l'algorithme de Newton pour la résolution de cette équation

---

Données : y, n le nombre de points de Gauss-Laplace à utiliser pour les calculs directs, ϵ une précision

Résultat : x_N qui approxime la solution de $\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x} \exp^{-t^2/2}\ dt = y$

début

$x_1 \longleftarrow 0$;$x_0 \longleftarrow x_1 + 2 \times \epsilon$;

tant que $Nx > 0$ *et* $|x_1 - x_0| > \epsilon$ **faire**

$x_0 \longleftarrow x_1$;

$x_1 \longleftarrow x_0 + \sqrt{2\pi}\ exp^{-x_0^2/2}(y - P(x_0, n))$;

$Nx = Nx - 1$

fin

si $Nx < 0$ **alors**

$x_1 \longleftarrow$ "échec de l'algorithme"

fin

fin

Question 9 Écrire la fonction maxima qui réalise cet algorithme

```
InvP(y,n,prec,Nx):=block([x1:0,x0:2*prec],
while (Nx>0 and abs(x1-x0) > prec) do
(x0:x1,x1:float(x0+sqrt(2*%pi)*exp(x0^2/2)*(y-P(x0,n))),Nx:Nx-1),
x1);
```

ou mieux

```
InvP(y,n,prec,Nx):=block([x1:0,x0:2*prec,setp],
if integerp(n) then setp:support_et_poids_de_gauss_legendre(n)
               else (setp:n,n:length(setp)),
while (Nx>0 and abs(x1-x0) > prec) do
(x0:x1,x1:float(x0+sqrt(2*%pi)*exp(x0^2/2)*(y-P(x0,setp))),Nx:Nx-1),
x1);
```

Question 10 Vérifier les fonctions et les approprier à un usage .... domestique.

1. Comparaison de $\mathbb{P}(x, n)$ avec les valeurs exactes

   ```
   plot2d(P(x,5)-integrate(exp(-t^2/2),t,-inf,x)/sqrt(2*%pi),[x,-5,5]);
   ```

2. Test de $\mathbb{P}^{-1}$

   ```
   tst(y):=P(InvP(y,3,10^(-6),20),3)-y;
   plot2d(tst,[y,10^-2,1-10^(-2)]);
   ```